

Object-Oriented Programming Language

05/18/2023

Due:05/22/2023 11:59p.m.

Lab Assignment 11

1. Create a class Triangle and create a class derived from the base class Triangle called Isosceles. This main code you should not change. The triangle() function should be a member function of the Triangle class. The isosceles() function on the other hand, is a member function in the Isosceles class.

```
#include <iostream>

using namespace std;

int main(){
    Isosceles isc;
    isc.isosceles();
    isc.description();
    isc.triangle();
    return 0;
}
```

Output should look like this

```
I am an isosceles triangle
In an isosceles triangle two sides are equal
I am a triangle
```

2. Design a program that utilizes concepts of inheritance, protected members, and virtual functions to simulate a hierarchy of animals in a zoo.

Create a base class called Animal, which includes a protected member variable named *name* and a virtual function named *sound()*. Derive two classes, Cat and Dog, from the Animal base class. Implement the sound() function in the Cat and Dog classes to display their respective sounds. Create a Zoo class, which contains a dynamic array (you can use a vector internally) to store (pointer to) Animal objects.

This main function you should not change.

```
int main() {
    Zoo zoo;
```

```

    Cat cat("Oscar");
    Dog dog("Buddy");

    zoo.addAnimal(&cat);
    zoo.addAnimal(&dog);

    zoo.makeAllSounds(); // output comes from this call

    return 0;
}

```

Your output should look like this.

```

The cat meows.
The dog barks.

```

- Write a program that uses the Fraction class to represent fractions. The class should have two private member variables, numer and denom, representing the numerator and denominator of the fraction, respectively. Implement a *print* function that returns the string representation of the fraction. Overload the input operator >> and the output operator << to read fractions from standard input and output fractions to standard output. This is the main code you should not change:

```

int main() {
    Fraction f1(2, 4);
    Fraction f2;

    cin >> f2;
    cout << "Fractions are: ";
    cout << f1 << ", ";
    cout << f2 << endl;
}

```

Your output should look like this.

```

77/95
Fractions are: 2/4, 77/95

```

```

15/7
Fractions are: 2/4, 15/7

```

4. Write a derived class `RFraction` that inherits from the `Fraction` class in the previous question. Override the *print* function in the `RFraction` class. The `RFraction` class should have an additional feature to reduce the fraction and output it. This main code you should not change.

```
int main() {  
    RFraction rf1(2, 4);  
    RFraction rf2;  
    cin >> rf2;  
    cout << "Reduced fractions are: ";  
    cout << rf1 << ", ";  
    cout << rf2 << endl;  
    return 0;  
}
```

Your output should look like this.

```
84/21  
Reduced fractions are: 1/2, 4/1
```

```
505/20  
Reduced fractions are: 1/2, 101/4
```

5. Write a base class `Shape` and derived classes `Triangle`, `Circle`, and `Rectangle`. The base class `Shape` has two virtual functions:

```
double getArea();  
string getClassName();
```

The `<<` operator should also be overloaded for the class `Shape` only as the second operand.

Please complete the necessary data members, constructors, and functions so that the following main function can produce the corresponding output. This main code you should not change.

```
int main(){  
    vector<Shape*> vs;  
    Rectangle r(10,20);  
    Circle c(10);  
    Triangle t(18,30,24);  
    vs.push_back(&r);  
    vs.push_back(&c);
```

```

        vs.push_back(&t);
        for(auto s : vs)
            cout << s;
    }

```

Your output should look like this.

```

Rectangle's area is 200
Circle's area is 314.15
Triangle's area is 216

```

6. Write a C++ program that includes a Shark class and a derived class BabyShark. The Shark class has an age member variable and implements a constructor, a print function, and a destructor. The BabyShark class extends the Shark class and adds a babyage member variable. Implement the necessary constructor, print function, and destructor for the BabyShark class. Also, Overload the output operator << to print a Shark object or its derived classes to the standard output. This main code you should not change.

```

int main() {
    cout << "-----a-----" << endl;
    BabyShark shark1(20);
    BabyShark shark2(6);
    vector<BabyShark> vec;
    vec.push_back(BabyShark(35));
    cout << "-----b-----" << endl;
    cout << shark1 << shark2 << vec[0];
    cout << "-----c-----" << endl;
}

```

Your input and output should look like this:

```

-----a-----
New Shark with age: 20
New BabyShark with age: 20
New Shark with age: 6
New BabyShark with age: 6
New Shark with age: 35
New BabyShark with age: 35
Age 35 BabyShark~~ Bye!
Age 35 Shark~~ Bye!
-----b-----

```

```

Age 20 Shark~~ BOO!BOO!BOO!
Age 20 BabyShark~~ DOO!DOO!DOO!
Age 6 Shark~~ BOO!BOO!BOO!
Age 6 BabyShark~~ DOO!DOO!DOO!
Age 35 Shark~~ BOO!BOO!BOO!
Age 35 BabyShark~~ DOO!DOO!DOO!
-----c-----
Age 35 BabyShark~~ Bye!
Age 35 Shark~~ Bye!
Age 6 BabyShark~~ Bye!
Age 6 Shark~~ Bye!
Age 20 BabyShark~~ Bye!
Age 20 Shark~~ Bye!

```

7. This is an extension to the previous problem. You are to add a virtual function in the two classes. You will practice dynamic binding. The following is the main code you are to use without changing.

```

int main() {
    cout << "-----a-----" << endl;
    Shark shark0(35);
    BabyShark shark1(20);
    BabyShark shark2(6);

    vector<Shark*> vec;
    vec.push_back(&shark0);
    vec.push_back(&shark1);
    vec.push_back(&shark2);
    cout << "-----b-----" << endl;

    for(auto& i : vec)
        i->dance();

    cout << "-----c-----" << endl;
}

```

The output:

```

-----a-----
New Shark with age: 35
New Shark with age: 20

```

```
New BabyShark with age: 20
New Shark with age: 6
New BabyShark with age: 6
-----b-----
This is Shark dancing!!!!
This is BabyShark dancing!!!!
This is BabyShark dancing!!!!
-----c-----
Age 6 BabyShark~~ Bye!
Age 6 Shark~~ Bye!
Age 20 BabyShark~~ Bye!
Age 20 Shark~~ Bye!
Age 35 Shark~~ Bye!
```